

RECOGNIZING SIGNED-GRAPHIC MATROIDS: CYLINDER FLIPS AND THE IMPORTANCE OF COLUMN SCALING

LISA SEUNG-YEON LEE

MAY 6, 2014

ABSTRACT. In this paper, we investigate the importance of column scaling in relating two signed-graphic representations of the same matroid. We used the Sage Mathematics software to generate many examples of signed-graphic matroids and their signed-graphic representations. Our examples show that column scaling is sometimes necessary in order to transform one signed-graphic representation into another; moreover, there exist many collections of signed-graphic representations that row-reduce to the same standard form. We also discuss an interesting matroid-preserving operation on a signed graph, which we call the cylinder flip, that relates certain pairs of signed-graphic representations of the same matroid.

CONTENTS

1. Introduction	2
1.1. Our approach	2
2. Definitions and notation	2
2.1. Signed graphs	3
2.2. Partial fields	3
2.3. Dyadic matroids	4
2.4. Signed-graphic matroids	4
2.5. Cylinder graph	5
3. Generating dyadic matroids	6
3.1. Unique \mathbb{D} -representation	6
3.2. 3-connectedness	7
4. Generating signed-graphic representations	8
4.1. Initial call to the recursive function	8
4.2. The recursive function	8
5. Matroid-preserving operations on a signed-graphic representation	13
5.1. Cylinder flip	13
5.2. Example	17
6. Restricting column scaling factors to ± 1	19
6.1. Preliminaries	19
6.2. Brylawski-Lucas Theorem	19
6.3. Restricting column scaling	21
7. Conclusion	21
Appendix A. Examples of signed-graphic representations	22
A.1. A matroid with some row-equivalent, some non-row-equivalent signed-graphic representations	22
A.2. A matroid with all row-equivalent signed-graphic representations	25
A.3. A matroid with all non-row-equivalent signed-graphic representations	28
Acknowledgements	29
References	29

1. INTRODUCTION

A class \mathcal{M} of matroids is *polynomial-time recognizable* if there is a polynomial $f(x)$ and an algorithm that determines after at most $f(n)$ rank evaluations, whether or not a fixed n -element matroid M is in \mathcal{M} . Seymour (1981a) proved that the class of binary matroids is not polynomial-time recognizable, although the class of graphic matroids is. Later, Truemper (1982b) proved that the class of regular matroids is polynomial-time recognizable. Geelen and Mayhew independently showed that if a signed-graphic matroid is given by a rank oracle, then it is not polynomial-time recognizable. Since every signed-graphic matroid is a dyadic matroid, it follows that the class of dyadic matroids is not polynomial-time recognizable.

A matrix A is *binet* if $[I \mid A]$ is row-equivalent to an incidence matrix of a signed graph. In his doctoral thesis [Mus10], Musitelli presents a first polynomial-time algorithm for recognizing binet matrices. In particular, his algorithm inputs a matrix A that is potentially binet, and tries to reverse-engineer (using only row operations) the matrix into a signed-graphic incidence matrix. If A is binet, then his algorithm finds a sequence of row operations bringing the matrix into a signed-graphic incidence matrix; otherwise, his algorithm determines that A is not binet.

However, we saw a number of concerns in Musitelli's work. Musitelli considers only row operations, and not column scaling, in order to recover a signed graph matrix from a binet one. Moreover, several different signed graph matrices can have the same row-reduced standard representation, but Musitelli does not mention which signed graph his algorithm returns, given a binet matrix.

This research was motivated by the desire to investigate the importance of column scaling in relating two signed-graphic representations of the same matroid.

1.1. Our approach. We used the Sage Mathematics software [S⁺09] to generate many examples of signed-graphic matroids and their signed-graphic representations. In particular, we first generated all non-isomorphic 3-connected uniquely \mathbb{D} -representable matroids of size up to 10 that could potentially be signed-graphic; then for each of these matroids, we computed all its non-isomorphic signed-graphic representations, and used Sage Notebook to draw the corresponding signed graphs. Our algorithms are discussed in Sections 3 and 4.

Using our Sage code, we were able to generate many interesting examples of signed graphs. Out of the 69 dyadic matroids we generated, 13 matroids had signed-graphic representations such that each pair of representations is row-equivalent; 39 had signed-graphic representations such that each pair is not row-equivalent; and the remaining 17 had signed-graphic representations such that some pairs are row-equivalent and some are not. Appendix A provides an example of each kind.

In Section 5 we discuss an interesting matroid-preserving operation on a signed graph, which we call the *cylinder flip*, that relates certain pairs of signed-graphic representations of the same matroid. Then in Section 6 we present the Brylawski-Lucas Theorem, and illustrate how restricting column scaling to the factors ± 1 weakens the theorem.

2. DEFINITIONS AND NOTATION

We assume familiarity with matroid theory and graph theory, and refer to [Oxl11] for any concepts and notation that remain undefined.

2.1. Signed graphs. The tuple $\Omega = (G, \Sigma)$, where $G = (V, E)$ is a graph and $\Sigma \subseteq E$, is a *signed graph*, the edges in Σ being the *negative* edges. The *sign* of a path or cycle in G is *positive* if there is an even number of negative edges in it, and *negative* otherwise.

2.1.1. Signed graph matrix. A signed graph is *oriented* when each end of each edge is given a direction, so that in a positive edge the ends are both directed from one endpoint to the other, and in a negative edge either both ends are directed outward, to their own vertices, or both are directed inward, away from their vertices. Note that an oriented signed graph is a bidirected graph.

Given an orientation of a signed graph Ω , its *incidence matrix* is a $|V| \times |E|$ matrix over $GF(3)$, with a row for each vertex and a column for each edge, whose ij th entry is

$$a_{ij} = \begin{cases} 0 & \text{if vertex } i \text{ and edge } j \text{ are not adjacent, or if edge } j \text{ is a positive loop in } \Omega \\ 1 & \text{if edge } j \text{ is oriented into vertex } i \\ -1 & \text{if edge } j \text{ is oriented out of vertex } i, \text{ or if edge } j \text{ is a negative loop incident with vertex } i \text{ in } \Omega \end{cases}$$

2.1.2. Binet matrices. A matrix A is called a *binet matrix* if there exist both an integral matrix M of full rank satisfying

$$\sum_{i=1}^n |M_{ij}| \leq 2 \quad \text{for any column index } j$$

and a basis B of it such that $M = [B \mid N]$ (up to column permutation) and $A = B^{-1}N$. In other words, A is binet if $[I \mid A]$ is row-equivalent to the incidence matrix of a bidirected graph.¹ Because a signed graph matrix is the incidence matrix of a bidirected graph, the signed graph matrices row-reduced into standard form are a subset of binet matrices.

2.2. Partial fields. A *partial field* is a pair $\mathbb{P} = (R, G)$, where R is a commutative ring, and $G \subseteq R^*$ is a subgroup such that $-1 \in G$. If $p \in G \cup \{0\}$, then we say p is an *element* of \mathbb{P} , and write $p \in \mathbb{P}$.

2.2.1. Matrix representation over a partial field. Let A be a matrix over R having r rows. Then A is a *weak \mathbb{P} -matrix* if, for each $r \times r$ submatrix D of A , we have $\det(D) \in G \cup \{0\}$.

A weak \mathbb{P} -matrix A is a *matrix representation* of a matroid M on elements E if for every r -subset $X \subseteq E$, the submatrix $A[X]$ of A (indexed by the elements in X) has the property that

$$\det(A[X]) \neq 0 \quad \Leftrightarrow \quad X \text{ is a basis of } M,$$

and there exists at least one r -subset $B \subseteq E$ such that $\det(A[B]) \neq 0$. If such a matrix representation exists for a matroid M , then we say M is *representable* over \mathbb{P} .

¹Musitelli's definition of the node-edge incidence matrix of a bidirected graph is different from ours in that he distinguishes between half-edges and loops. For our purposes, this distinction is irrelevant and a mere semantic difference.

2.3. Dyadic matroids. The *dyadic partial field* is

$$\mathbb{D} := (\mathbb{Q}, \{\pm 2^k : k \in \mathbb{Z}\}).$$

A *dyadic matrix* is a matrix over \mathbb{Q} all of whose nonzero subdeterminants are in $\{\pm 2^i : i \in \mathbb{Z}\}$. A matroid is *dyadic* if it is representable over \mathbb{D} . The following theorem provides equivalent definitions for a dyadic matroid:

Theorem 2.3.1 (Whittle, 1997). *The following are equivalent for a matroid M .*

- (i) M is dyadic.
- (ii) M is representable over $GF(3)$ and $GF(5)$.
- (iii) M is representable over $GF(p)$ for all odd primes p .
- (iv) M is representable over $GF(3)$ and \mathbb{Q} .
- (v) M is representable over $GF(3)$ and \mathbb{R} .
- (vi) M is representable over $GF(3)$ and $GF(q)$ where q is an odd prime power such that $q \equiv 2 \pmod{3}$.

Dyadic matroids are a natural analogue of regular matroids. Notably, Seymour's Decomposition Theorem for totally unimodular matrices characterizes the regular matroids as the 3-sums of graphic matroids, cographic matroids, and a certain 10-element matroid. In a similar spirit, as graphic matroids are to regular matroids, the signed-graphic matroids are a fundamental subclass of dyadic matroids.

2.4. Signed-graphic matroids. A matroid is *signed-graphic* if it can be represented by an incidence matrix of a signed graph. Since the incidence matrix of a signed graph is a matrix over $GF(3)$ whose columns have at most 2 nonzero entries each, it follows that every signed-graphic matroid is dyadic. We say two signed-graphic representations are *row-equivalent* if one can be obtained from the other via a sequence of elementary row operations.

2.4.1. Circuits and bases. The circuits of a signed graph are precisely the positive cycles, pairs of negative cycles that meet exactly in a vertex, and pairs of disjoint negative cycles, together with any path connecting them.

By a slight abuse of definition, we call an independent set of edges a *forest* of the signed graph. Since every dependent set has a connected component containing a positive cycle or two negative cycles, the forests of a signed graph are precisely the subgraphs in which every connected component contains at most one negative cycle (and no positive cycles). A basis of the signed-graphic matroid corresponds to a *spanning forest* of the signed graph.

2.4.2. Resigning edges of a signed-graphic representation. For two sets S, T we define

$$S\Delta T := \{s \in S \mid s \notin T\} \cup \{t \in T \mid t \notin S\}.$$

If $G = (V, E)$ is a graph, and $S \subseteq V$, then $\delta(S) := \{e = uv \in E \mid u \in S, v \notin S\}$.

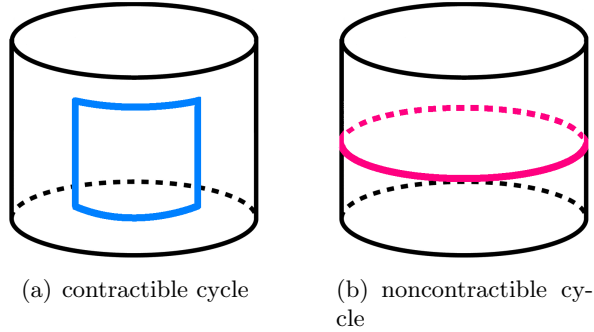
Let $\Omega = (G, \Sigma)$ be a signed graph. For a vertex v , we can *resign* Ω around v by replacing Σ by $\Sigma\delta(v)$. Resigning edges across a vertex set corresponds to row scaling by -1, and hence preserves the corresponding matroid $M(\Omega)$. We state this in a lemma:

Lemma 2.4.1. *Let Ω be the signed-graphic representation of a matroid M . For a vertex $v \in V(\Omega)$, let Ω' be obtained by resigning Ω around v . Then $M(\Omega) \cong M(\Omega')$.*

A pair of vertices $\{u, v\}$ is called a *blocking pair* if there exists a resigning of the edges such that every negative cycle meets u or v .

2.5. Cylinder graph.

2.5.1. *Contractible edges on a cylinder.* Given a graph G embedded on the cylinder, let C be a cycle of G . We say C is *contractible* if we can repeatedly contract the edges of C so that C becomes a single vertex, and *noncontractible* otherwise.



2.5.2. *Cylinder graph.* We define the *cylinder graph* of a signed graph $\Omega = (G, \Sigma)$ in the following way. Suppose Ω has two blocking pairs $\{s_1, s_2\}$ and $\{t_1, t_2\}$ such that $\{s_1, s_2, t_1, t_2\}$ form a 4-vertex cut. Because $\{t_1, t_2\}$ is a blocking pair, there exists a resigning of the edges such that every negative cycle meets t_1 or t_2 . We use this resigning to embed Ω on a cylinder such that all positive cycles are contractible and all the negative cycles are noncontractible.

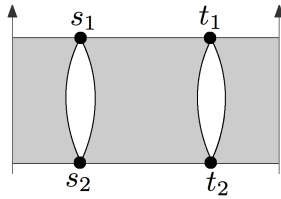
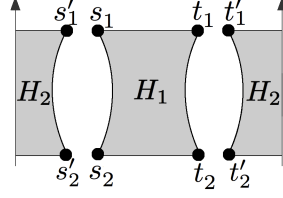


FIGURE 1. Ω embedded on a cylinder, where the edges only cross inside the gray area. The cylinder has been cut open along the arrows.

If we split each of the blocking pair vertices into two (i.e., s_i into s_i and s'_i , and t_i into t_i and t'_i) as shown in the figure below, then Ω becomes divided into two components (each of which is not necessarily connected), call them H_1 and H_2 .

FIGURE 2. Ω after splitting each of the blocking pair vertices into two.

Now consider the original embedding of Ω on the cylinder, before the blocking pair vertices were split. Suppose there exists a resigning Σ' of the edges such that every negative edge is incident with either t_1 or t_2 , and the negative edges are either all contained in H_1 or all contained in H_2 . Then such embedding of the signed graph (G, Σ') on the cylinder is called a *cylinder graph on $\{s_1, s_2; t_1, t_2\}$* .

We note that our definition of cylinder graphs is slightly different from the definition in [PZ07]; for example, our definition does not require planarity.

3. GENERATING DYADIC MATROIDS

To generate all non-isomorphic 3-connected uniquely \mathbb{D} -representable matroids of size up to k that could be potentially signed-graphic, our algorithm incrementally computes the simple dyadic extensions and cosimple dyadic coextensions of each $M \in \{F_7^-, (F_7^-)^*\}$. More explicitly:

```

Generate-Matroids {
   $\mathcal{M} \leftarrow \{F_7^-, (F_7^-)^*\}$ 
  for  $i$  from 1 to  $(k - 7)$ :
    for each  $M$  in  $\mathcal{M}$ : {
       $\mathcal{N} \leftarrow$  all simple dyadic single-element extensions and cosimple dyadic
        single-element coextensions of  $M$ 
      for each  $N \in \mathcal{N}$ :
        Add  $N$  to  $\mathcal{M}$  if it is not isomorphic to any previously added matroid.
    }
  return  $\mathcal{M}$ 
}

```

In the following sections, we explain the correctness of our algorithm and why these matroids are desirable for our purposes.

3.1. Unique \mathbb{D} -representation. Since we are interested in the row operations and column scalings required to get from one signed-graphic representation to another of the same matroid, we would like to generate only the matroids that have unique dyadic representations.

By [Zwa09, 6.3.4], P_8 , F_7^- , and $(F_7^-)^*$ are uniquely \mathbb{D} -representable; and moreover, they are a stabilizer for \mathbb{D} . So the simple extensions and cosimple co-extensions over \mathbb{D} of P_8 , F_7^- , and $(F_7^-)^*$ are uniquely \mathbb{D} -representable, by the following proposition ([Oxl11, 14.8.2]):

Proposition 3.1.1. *Let \mathbb{F} be a field and N be an \mathbb{F} -stabilizer for the class \mathcal{M} of \mathbb{F} -representable matroids. If N is uniquely representable over \mathbb{F} , then so is every 3-connected matroid in \mathcal{M} that has an N -minor.*

However, we are not interested in the extensions and coextensions of P_8 because matroids with a P_8 minor are not signed-graphic.

3.2. 3-connectedness. Since every 2-connected matroid can be written in terms of 2-sums of 3-connected matroids ([Oxl11, Cunningham and Edmonds 1980, Seymour 1981b]), it is natural to investigate only the 3-connected matroids.

Lemma 3.2.1. *If $M \setminus e$ is 3-connected and M is not 3-connected, then e must be a parallel pair or a loop.*

Proof. Let (A, B) be a 2-separation of M with $e \in B$. Then by definition of 2-separation,

$$\begin{aligned} r(A) + r(B) - r(A \cup B) &\leq 1 \\ |A|, |B| &\geq 2. \end{aligned}$$

If we delete e , then $r(B \setminus e) - r(A \cup B \setminus e)$ will either stay the same or decrease by 1, and so

$$r(A) + r(B \setminus e) - r(A \cup B \setminus e) \leq 1.$$

But because $M \setminus e$ is 3-connected, $(A, B \setminus e)$ cannot be a 2-separation of $M \setminus e$. So $|B| = 2$, i.e., $B = \{e, f\}$ for some element f . If $\{e, f\}$ is a series pair, then $M \setminus e$ will have 1-separation, contradicting the assumption that $M \setminus e$ is 3-connected. Therefore, e must be a parallel pair or a loop. \square

Lemma 3.2.2. *If M/e is 3-connected and M is not 3-connected, then e must be a series pair or a coloop.*

Proof. Since 3-connectedness is closed under duality, and contraction in M corresponds to deletion in M^* , applying Lemma 3.2.1 to M^* concludes the proof. \square

Since the algorithm computes simple extensions and cosimple coextensions of F_7^- and $(F_7^-)^*$, the algorithm never adds any parallel pairs, loops, series pairs, or coloops to a 3-connected matroid. So it follows from the above lemmas that the algorithm produces 3-connected matroids.

More generally, consider the following important result in matroid theory (Seymour 1980b):

Theorem 3.2.3 (Splitter Theorem). *Let N be a non-empty, connected, simple, cosimple minor of a 3-connected matroid M . Suppose that N is neither a wheel nor a whirl. Then either $M = N$, or M has a connected, simple, cosimple minor M_1 such that some single-element deletion or some single-element contraction of M_1 is isomorphic to N . Moreover, if N is 3-connected, so too is M_1 .*

Wheels and whirls are an exception to the Splitter theorem, but this is not a problem in our case since there are no wheels or whirls in matroids with F_7^- , $(F_7^-)^*$, or P_8 as a minor.

Corollary 3.2.4. *Let M and N be 3-connected matroids such that N is a minor of M with at least four elements, and N is neither a wheel nor a whirl. Then there is a sequence M_0, M_1, \dots, M_n of 3-connected matroids with $M_0 \cong N$ and $M_n = M$ such that M_i is a single-element deletion or a single-element contraction of M_{i+1} for all $i \in \{0, 1, \dots, n-1\}$.*

Starting with $M_0 \in \{F_7^-, (F_7^-)^*\}$, our algorithm generates a set containing all sequences M_0, M_1, \dots, M_n of 3-connected matroids such that M_{j+1} is a single-element extension or coextension of M_j .

4. GENERATING SIGNED-GRAPHIC REPRESENTATIONS

To generate all signed-graphic representations of a given dyadic matroid, we implemented Prof. Rudi Pendavingh's recursive function `graphic_rec` (originally written in C++) in Sage. The following nonrecursive pseudocode provides a simplified overview of the algorithmic approach, although the algorithm performs some clever optimizations to improve the running time.

```

graphic_rec (input: dyadic matroid  $M$ ) {
   $\mathcal{A} \leftarrow \emptyset$ 
   $R \leftarrow$  matrix representation of  $M$ 
   $X \leftarrow$  all dyadic extension columns of  $M$ 
  for each full-rank  $r$ -subset  $S \subseteq X$ : {
    Row-reduce  $[R \mid S] \rightarrow [R' \mid I]$ 
    if  $R'$  has  $\leq 2$  nonzeros per column:
      Add  $R'$  to  $\mathcal{A}$ 
  }
  Return  $\mathcal{A}$ 
}

```

If X is the set of all dyadic extension columns of a rank- r matroid M , then the columns of X correspond to a finite number of points in the ambient space. The algorithm tries to find an r -subset of these points $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_r \in X$ such that $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_r\}$ form a basis and every original element of the matroid is spanned by at most two elements from this basis. If such a basis exists, then the matroid can be represented as $[I \mid A]$, where the columns of A correspond to the original elements of the matroid, and each column of A contains at most 2 nonzero entries. Then A is a signed-graphic representation of M .

In each call to the recursive function, the algorithm grabs an appropriate new element \vec{x} from X and turns it into a basis element, then checks if the current (incomplete) set of basis elements can be extended into a basis with the desired properties. Essentially, the algorithm adds one edge at a time to grow a spanning forest, similar to Prim's greedy algorithm for growing a minimum spanning tree of a connected graph.

4.1. Initial call to the recursive function. Given a rank- r dyadic matroid M on n elements, let R be a dyadic representation of M , and X the set of all dyadic extension columns of M . The algorithm calls the recursive function `graphic_rec` (described in the following section) with the initial parameters $k = 1$, R , and X .

4.2. The recursive function. The `graphic_rec` function inputs an integer $k \geq 1$, and two matrices R and X of the form

$$R = \left(\begin{array}{c|c} I_{k-1} & \\ \hline 0 & A \end{array} \right) \quad X = \left(\begin{array}{c|c} I_{k-1} & * \\ \hline 0 & \geq 1 \text{ nonzeros per column} \end{array} \right)$$

where the submatrix A satisfies the following property for $N = k - 1$:

Property 4.2.1. If a column has more than 2 nonzero entries in the first N entries, then $\exists \ell \geq N + 1$ such that the ℓ th entry of that column is nonzero.

We require this condition on R for the following reason: If a column of A has more than 2 nonzero entries in the already processed rows, then a future pivot is needed to repair that column so that it has ≤ 2 nonzero entries. If the property is not satisfied, then the column cannot be repaired and hence R cannot be signed-graphic.

The first $k - 1$ columns of R and of X correspond to the incomplete set of basis elements that the algorithm has already found prior to the current recursive call. Furthermore, this incomplete set of basis elements corresponds to a forest F , which we would like to grow into a spanning forest. In choosing the next element to add to the matroid, the algorithm considers three cases: it can grow an existing partial component of F into a larger one, add a negative loop, or declare the existing components to be done and start a new component.

4.2.1. Growing an existing partial component. Suppose there exists a column $\vec{x} \in X$ such that it has 1 nonzero entry among the first k rows, and exactly one nonzero entry among the remaining $r - k$ rows. Then we may try to grow an existing component of the forest F . Let $t < k$ and $s \geq k$ be the unique indices such that $b := \vec{x}[t]$ and $a := \vec{x}[s]$ are nonzero.

If $R' := R \cup \{\vec{x}\}$ is obtained by inserting \vec{x} between the $(k - 1)$ th and k th columns in R , then

$$R' = \left(\begin{array}{c|c} I_{k-1} & \vec{x} \\ \hline 0 & A \end{array} \right) = \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & b & \vec{y} \\ \hline & & \\ \hline 0 \cdots 0 & a & \vec{z} \end{array} \right)$$

where \vec{y} (resp. \vec{z}) is the t th (resp. s th) row of A . To turn \vec{x} into a basis element, the algorithm pivots over $\vec{x}[s]$, then swaps the k th row and s th row of R :

$$\begin{array}{ccc} \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & b & \vec{y} \\ \hline & & \\ \hline 0 \cdots 0 & a & \vec{z} \end{array} \right) & \rightarrow & \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & 0 & \vec{y} - ba^{-1}\vec{z} \\ \hline & & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{z} \end{array} \right) & \rightarrow & \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & 0 & \vec{y} - ba^{-1}\vec{z} \\ \hline & & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{z} \end{array} \right) \\ (1) R' & & (2) \text{Pivot over } \vec{x}[s] = R'_{sk} & & (3) \text{Swap rows } k \text{ and } s \end{array}$$

Now R' is of the form

$$R' = \left(\begin{array}{c|c} I_k & \\ \hline 0 & A' \end{array} \right)$$

The algorithm then checks whether A' satisfies property 4.2.1 for $N = k$. If A' does not satisfy the property, then R' can no longer be turned into a signed-graphic representation, so the algorithm prunes search by returning the current recursive function call.

Otherwise if $k = r$, then property 4.2.1 guarantees that $R' = [I \mid A]$ where every column of A has ≤ 2 nonzero entries, which implies that the representation is signed-graphic. So the algorithm stores A in \mathcal{A} and returns the current function call.

Otherwise if $k < r$, then the algorithm prepares for another recursive call. Without loss of generality, we may assume that \vec{x} is the k th column of X , i.e.,

$$X = \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & & * \\ \hline & \vec{x} & \\ \hline 0 & & \geq 1 \text{ nonzeros per column} \end{array} \right) = \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & b & \vec{u} \\ \hline & & \\ \hline 0 \cdots 0 & a & \vec{w} \end{array} \right)$$

First, the algorithm pivots X over $\vec{x}[s]$, then swaps the k th and s th row of X :

$$\begin{array}{ccc} \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & b & \vec{u} \\ \hline & & \\ \hline 0 \cdots 0 & a & \vec{w} \end{array} \right) & \rightarrow & \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline \vec{e}_t & 0 & \vec{u} - ba^{-1}\vec{w} \\ \hline & & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{w} \end{array} \right) & \rightarrow & \begin{array}{c} t \\ k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & 0 & \vec{u} - ba^{-1}\vec{w} \\ \hline & & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{w} \end{array} \right) \\ \text{(1) } X & & \text{(2) Pivot over } \vec{x}[s] = X_{sk} & & \text{(3) Swap rows } k \text{ and } s \end{array}$$

Then the algorithm throws away the columns in X which do not have a nonzero entry below the k th row (i.e., the columns which are entirely spanned by $\{e_1, e_2, \dots, e_k\}$). Finally, the algorithm recursively calls `graphic_rec` with the parameters $k + 1$, R' , and X .

4.2.2. Adding a negative loop. Suppose there exists a column $\vec{x} \in X$ such that its only nonzero entry is on or below the k th row. Then we may try to add a negative loop to the forest F . Let $s \geq k$ be the unique index such that $a := \vec{x}[s]$ is nonzero.

If $R' := R \cup \{\vec{x}\}$ is obtained by inserting \vec{x} between the $(k-1)$ th and k th columns in R , then

$$R' = \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & & \\ \hline & \vec{x} & \\ \hline 0 & & A \end{array} \right) = \begin{array}{c} k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline 0 \cdots 0 & a & \vec{z} \\ \hline & 0 & \end{array} \right)$$

where \vec{z} is the s th row of A . To turn \vec{x} into a basis element, the algorithm pivots over $\vec{x}[s]$, then swaps the k th row and s th row of R :

$$\begin{array}{ccc} \begin{array}{c} k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline 0 \cdots 0 & a & \vec{z} \\ \hline & 0 & \end{array} \right) & \rightarrow & \begin{array}{c} k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{z} \\ \hline & 0 & \end{array} \right) & \rightarrow & \begin{array}{c} k \\ s \end{array} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline 0 \cdots 0 & 1 & a^{-1}\vec{z} \\ \hline & 0 & \end{array} \right) \\ (1) R' & (2) \text{Pivot over } \vec{x}[s] = R'_{sk} & (3) \text{Swap rows } k \text{ and } s \end{array}$$

Now R' is of the form

$$R' = \left(\begin{array}{c|c} I_k & \\ \hline 0 & A' \end{array} \right)$$

The algorithm then checks whether A' satisfies property 4.2.1 for $N = k$. If A' does not satisfy the property, then R' can no longer be turned into a signed-graphic representation, so the algorithm prunes search by returning the current recursive function call.

Otherwise if $k = r$, then property 4.2.1 guarantees that $R' = [I \mid A]$ where every column of A has ≤ 2 nonzero entries, which implies that the representation is signed-graphic. So the algorithm stores A in \mathcal{A} and returns the current function call.

Otherwise if $k < r$, then the algorithm prepares for another recursive call. Similar to the previous case, the algorithm pivots X over $\vec{x}[s]$, swaps the k th and s th row of X , then throws away the columns in X which do not have a nonzero entry below the k th row. Finally, the algorithm recursively calls `graphic_rec` with the parameters $k+1$, R' , and X .

4.2.3. Starting a new component. Suppose we declare the existing components to be done, i.e., we have exhausted possibilities for growing already existing components and would like to start a new component. Then the algorithm throws away the columns in X that have a nonzero entry in the first $k-1$ rows (i.e., the columns that connect to already existing components), and arbitrary

chooses one of the remaining columns $\vec{x} \in X$ that has a nonzero entry on or below the k th row. Let $s \geq k$ be the smallest integer such that $a := \vec{x}[s] \neq 0$.

If $R' := R \cup \{\vec{x}\}$ is obtained by inserting \vec{x} between the $(k-1)$ th and k th columns in R , then

$$R' = \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & & \\ \hline & \vec{x} & \\ \hline & & A \\ \hline 0 & & \end{array} \right) = \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline & & \\ \hline k & & \\ s & 0 \cdots 0 & a \quad \vec{z} \\ \hline & \vec{b} & B \end{array} \right)$$

where \vec{z} is the s th row of A , B is the $(r-s) \times n$ submatrix of A below \vec{z} , and \vec{b} is the $(r-s)$ -subvector of \vec{x} below $\vec{x}[s]$. To turn \vec{x} into a basis element, the algorithm pivots over $\vec{x}[s]$, then swaps the k th row and s th row of R :

$$\begin{array}{ccc} \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline & & \\ \hline k & & \\ s & 0 \cdots 0 & a \quad \vec{z} \\ \hline & \vec{b} & B \end{array} \right) & \rightarrow & \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline & & \\ \hline k & & \\ s & 0 \cdots 0 & 1 \quad a^{-1}\vec{z} \\ \hline & 0 & B - a^{-1}\vec{b}\vec{z} \end{array} \right) & \rightarrow & \left(\begin{array}{c|c|c} & k & \\ \hline I_{k-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} & \\ \hline & & \\ \hline k & 0 \cdots 0 & 1 \quad a^{-1}\vec{z} \\ \hline s & & 0 \quad B - a^{-1}\vec{b}\vec{z} \end{array} \right) \\ (1) R' & (2) \text{ Pivot over } \vec{x}[s] = R'_{sk} & (3) \text{ Swap rows } k \text{ and } s \end{array}$$

Now R' is of the form

$$R' = \left(\begin{array}{c|c} I_k & \\ \hline 0 & A' \end{array} \right)$$

The algorithm then checks whether A' satisfies property 4.2.1 for $N = k$. If A' does not satisfy the property, then R' can no longer be turned into a signed-graphic representation, so the algorithm prunes search by returning the current recursive function call.

Otherwise if $k = r$, then property 4.2.1 guarantees that $R' = [I \mid A]$ where every column of A has ≤ 2 nonzero entries, which implies that the representation is signed-graphic. So the algorithm stores A in \mathcal{A} and returns the current function call.

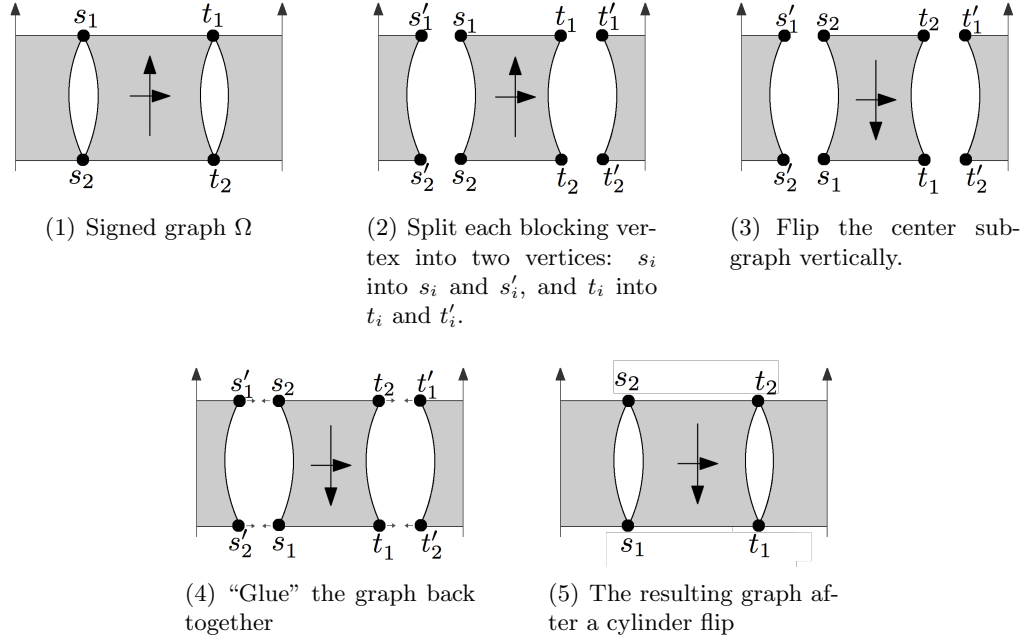
Otherwise if $k < r$, then the algorithm prepares for another recursive call. Similar to the previous cases, the algorithm pivots X over $\vec{x}[s]$, swaps the k th and s th row of X , then throws away the columns in X which do not have a nonzero entry below the k th row. Finally, the algorithm recursively calls `graphic_rec` with the parameters $k+1$, R' , and X .

5. MATROID-PRESERVING OPERATIONS ON A SIGNED-GRAPHIC REPRESENTATION

For certain pairs of signed-graphic representations A, B of the same matroid, we can get from A to B via a sequence of matroid-preserving operations. One such operation is the resigning of edges across a vertex set (see Lemma 2.4.1). We present another matroid-preserving operation called the *cylinder flip*, which is reminiscent of the Whitney flip [Piv11, Ch. 5]. In section 5.2, we present an example where two signed-graphic representations of the same matroid are related via a sequence of resigning edges and cylinder flips.

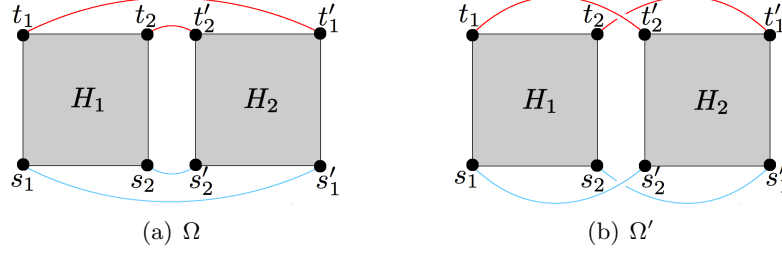
5.1. Cylinder flip.

Definition. For a given cylinder graph Ω on the blocking pair vertices $\{s_1, s_2; t_1, t_2\}$, the following sequence of operations on Ω is called a *cylinder flip over $\{s_1, s_2; t_1, t_2\}$* .


 FIGURE 3. A cylinder flip over $\{s_1, s_2; t_1, t_2\}$.

To show why the cylinder flip preserves a matroid, we first prove a lemma:

Lemma 5.1.1. *Let H_1 and H_2 be two signed graphs containing only positive edges. Let $s_1, s_2, t_1, t_2 \in V(H_1)$ and $s'_1, s'_2, t'_1, t'_2 \in V(H_2)$. Let Ω (resp. Ω') be the signed graph obtained by adding two positive edges $\{s_1, s'_1\}, \{s_2, s'_2\}$ (resp. $\{s_1, s'_2\}, \{s_2, s'_1\}$) and two negative edges $\{t_1, t'_1\}, \{t_2, t'_2\}$ (resp. $\{t_1, t'_2\}, \{t_2, t'_1\}$) between H_1 and H_2 . Then the corresponding matroids $M(\Omega)$ and $M(\Omega')$ of these signed graphs are equivalent.*

FIGURE 4. The signed graphs Ω and Ω' of Lemma 5.1.1.

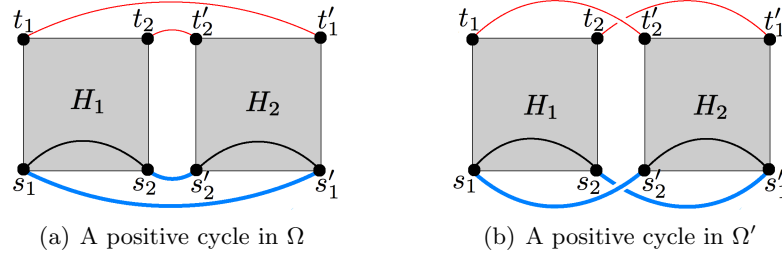
Proof. We will prove that if $X \subseteq E(\Omega)$ is a circuit of Ω , then the corresponding set of edges in Ω' is also a circuit of Ω' . The other direction follows by symmetry.

If X is contained entirely in H_1 or H_2 , then since Ω and Ω' have equivalent subgraphs H_1 and H_2 , X is a circuit in Ω if and only if it is a circuit in Ω' . So we may assume that $X \cap H_1 \neq \emptyset$ and $X \cap H_2 \neq \emptyset$.

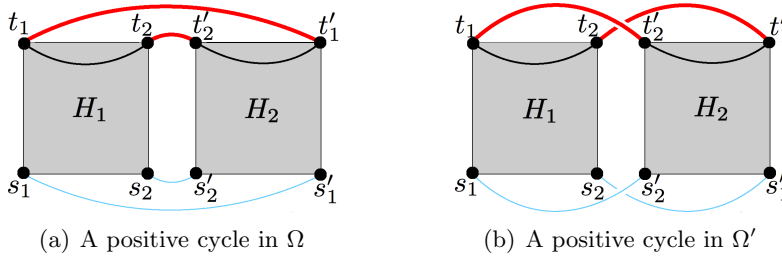
If $X \subseteq E(\Omega)$ is a circuit of Ω , then it is either a positive cycle, two negative cycles meeting at a point, or two negative cycles connected by a path.

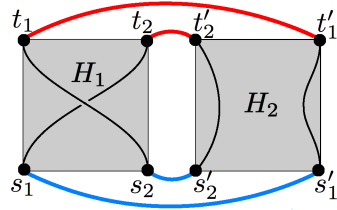
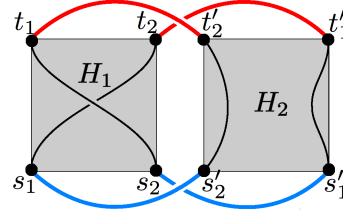
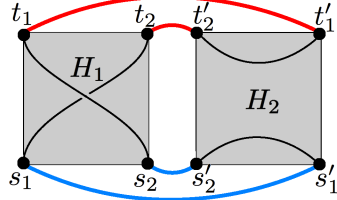
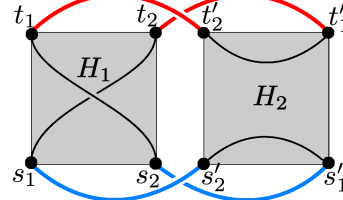
Case 1. Suppose X is a positive cycle, i.e., X contains an even number of negative edges. Since there are only two negative edges, X either contains 0 or 2 negative edges.

If X contains no negative edges, X must be a union of paths $P_1 \cup \{s_1, s'_1\} \cup P_2 \cup \{s_2, s'_2\}$ where P_1 (resp. P_2) is a path contained entirely in H_1 (resp. H_2). Then the same set of edges in $P_1 \cup \{s_1, s'_2\} \cup P_2 \cup \{s_2, s'_1\}$ form a positive cycle in Ω' :



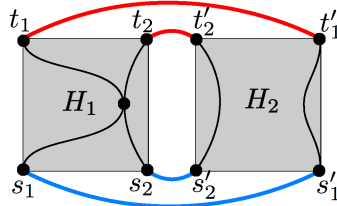
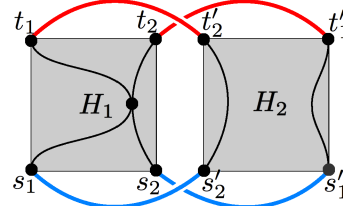
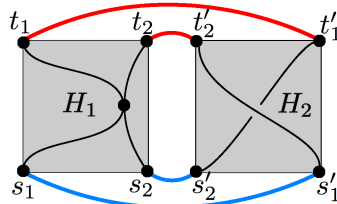
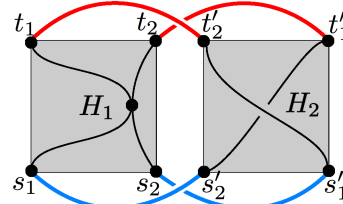
If X contains two negative edges, then the following figures enumerate all possible scenarios, up to symmetry. In all scenarios, the same set of edges also correspond to a positive cycle in Ω' :



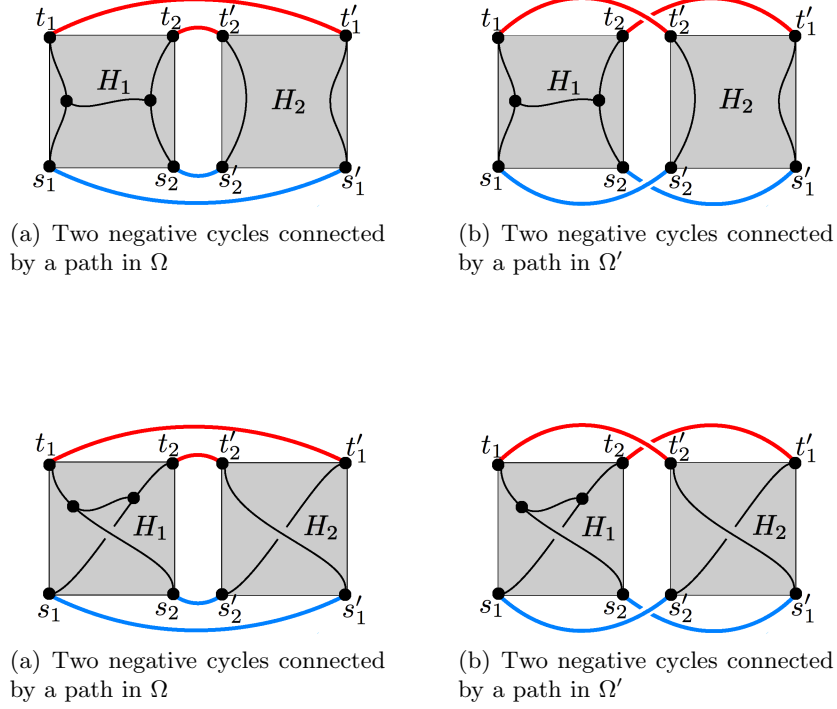

 (a) A positive cycle in Ω

 (b) A positive cycle in Ω'

 (a) A positive cycle in Ω

 (b) A positive cycle in Ω'

Case 2. Suppose X contains two negative cycles in Ω . Since Ω contains only two negative edges $\{t_2, t'_2\}$ and $\{t_1, t'_1\}$, each negative cycle must contain one of these negative edges. Moreover, since there are only two other edges $\{s_1, s'_1\}$ and $\{s_2, s'_2\}$ connecting H_1 and H_2 , each negative cycle must use one of these edges in order to close the walk.

Suppose the two negative cycles meet at a vertex v in Ω . Without loss of generality, we may assume $v \in H_1$. Then the following figures enumerate all possible scenarios, up to symmetry. In all scenarios, the same set of edges also correspond to two negative cycles meeting at a point in Ω' :


 (a) Two negative cycles meeting in a point in Ω

 (b) Two negative cycles meeting in a point in Ω'

 (a) Two negative cycles meeting in a point in Ω

 (b) Two negative cycles meeting in a point in Ω'

Otherwise, suppose the two negative cycles are joined by a path P in Ω . Without loss of generality, we may assume $P \subseteq H_1$. Then the same set of edges also correspond to two negative cycles joined by a path in Ω' :



We have shown that any circuit $X \subseteq E(\Omega)$ of Ω is also a circuit in Ω' . The proof of the other direction follows by symmetry. Therefore, we conclude that $M(\Omega) = M(\Omega')$. \square

Theorem 5.1.2 (Cylinder flip). *Let Ω be a cylinder graph on the blocking pair vertices $\{s_1, s_2; t_1, t_2\}$. If Ω' is obtained from Ω via a cylinder flip over $\{s_1, s_2; t_1, t_2\}$, then $M(\Omega) = M(\Omega')$.*

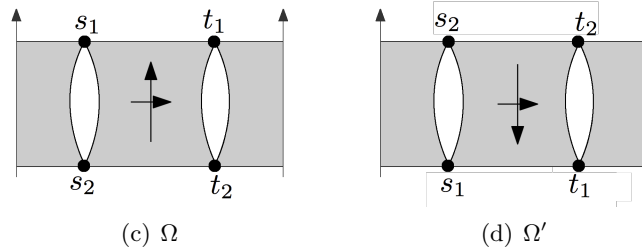
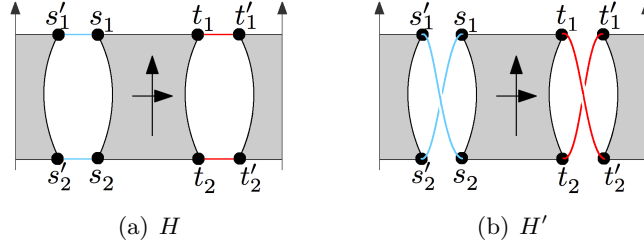


FIGURE 5. The signed graphs Ω and Ω' of Cor. 5.1.2.

Proof. By Lemma 5.1.1, the matroids of the following signed graphs H and H' are equivalent:



Resigning all edges incident with t_1 and t_2 , then contracting the edges $\{s_1, s'_1\}$, $\{s_2, s'_2\}$, $\{t_1, t'_1\}$, and $\{t_2, t'_2\}$ in H gives Ω . Similarly, resigning all edges incident with t_1 and t_2 , then contracting the edges $\{s_1, s'_2\}$, $\{s_2, s'_1\}$, $\{t_1, t'_2\}$, and $\{t_2, t'_1\}$ in H' gives Ω' . Since contraction in the signed graph corresponds to the abstract matroid contraction, we conclude that $M(\Omega) = M(\Omega')$. \square

5.1.1. *Degenerate cases.* With a little more work, we can also prove Theorem 5.1.2 for the degenerate cases, e.g. when $s_1 = t_1$.

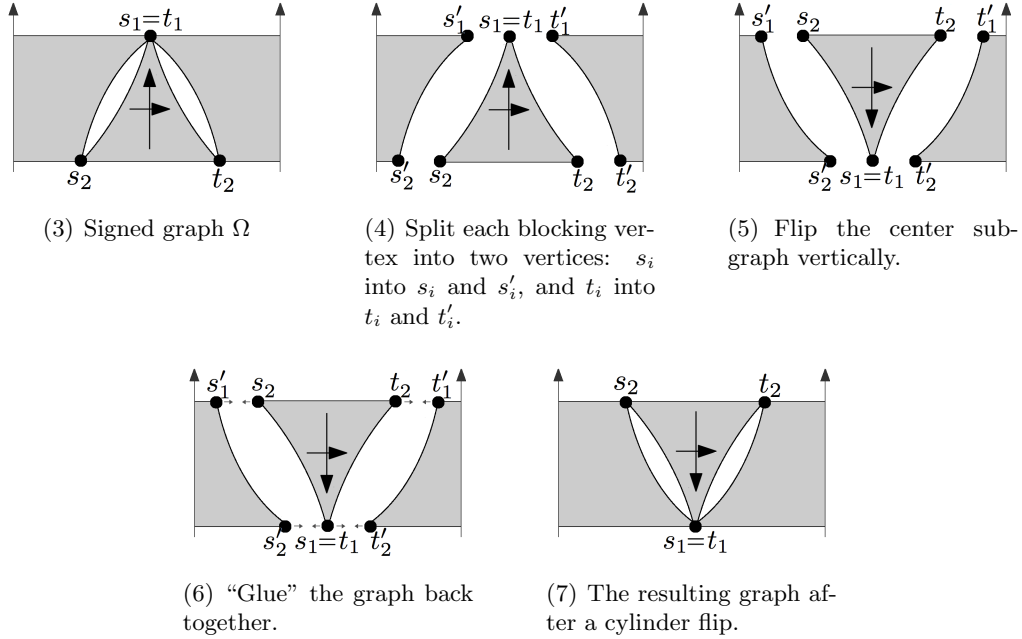
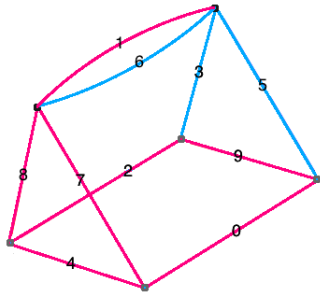


FIGURE 6. A cylinder flip in the degenerate case when $s_1 = t_1$.

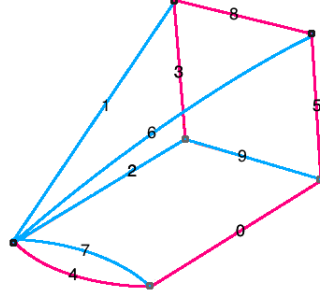
5.2. **Example.** Consider the dyadic matroid M of rank 6 on 10 elements whose \mathbb{D} -representation is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 1 \end{pmatrix}.$$

Two of its signed-graphic representations are given below²:



$$\Omega = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & \end{pmatrix}$$



$$\Omega' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The cylinder flip can be used to get from Ω to Ω' , as illustrated below.

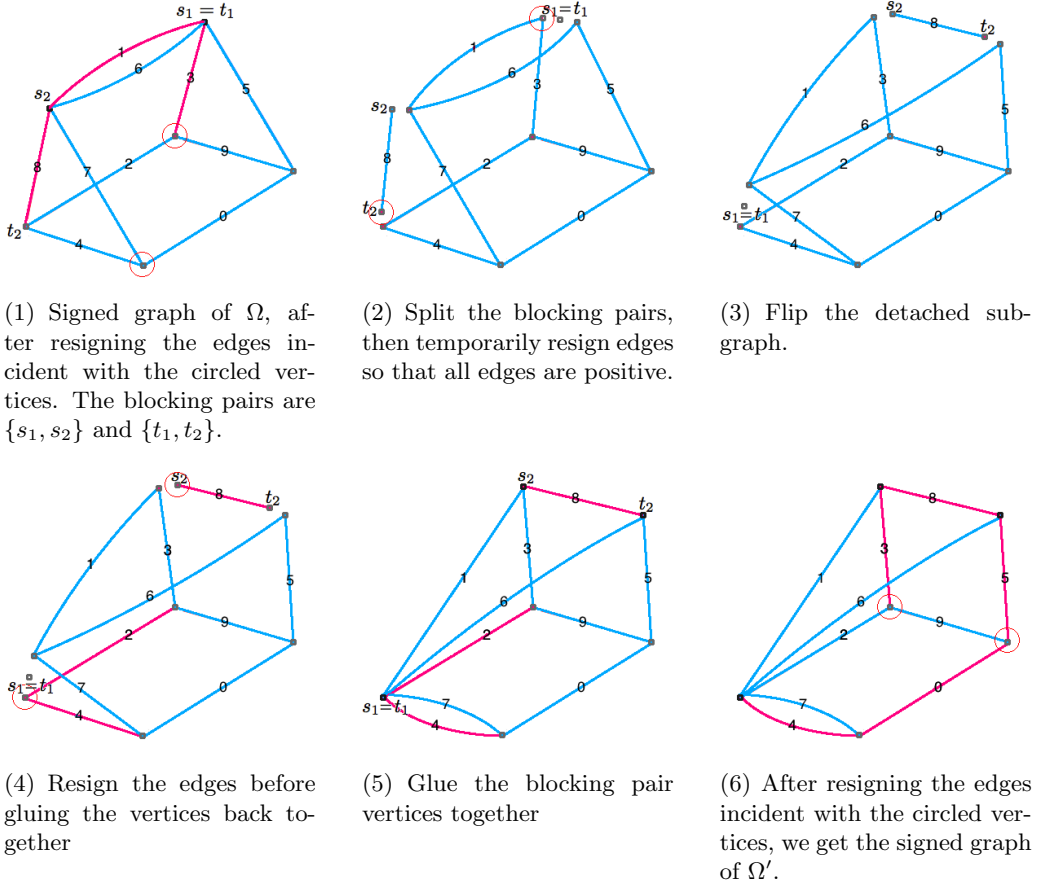


FIGURE 7. A sequence of matroid-preserving operations to get from Ω to Ω' .

²A complete list of its signed-graphic representations can be found in Appendix A.2.

6. RESTRICTING COLUMN SCALING FACTORS TO ± 1

If M is a rank- r matroid with $r \geq 1$, then every matrix that represents M over a field \mathbb{F} is projectively equivalent to a standard representative matrix $[I_r \mid D]$ for M . If we allow row and column scalings, then Theorem 6.2.1 ([Oxl11, 6.4.7]) shows that we can transform certain entries of D_1 into any nonzero elements $\theta_1, \theta_2, \dots, \theta_k \in \mathbb{F}$ that we want. On the other hand, if we restrict column scaling factors to ± 1 , then Lemma 6.3.1 shows that our choices of θ_i 's become very limited. Before stating Theorem 6.2.1, we first require some preliminaries.

6.1. Preliminaries. Given a standard representation $[I_r \mid D]$ for a matroid M , let its columns be labeled, in order, e_1, e_2, \dots, e_n . If B is the basis $\{e_1, e_2, \dots, e_r\}$, then it is natural to label the rows of $[I_r \mid D]$ by e_1, e_2, \dots, e_r . Hence D has its rows labeled by e_1, e_2, \dots, e_r and its columns labeled by $e_{r+1}, e_{r+2}, \dots, e_n$.

The B -fundamental-circuit incidence matrix of M , which we denote by $D^\#$, is the matrix obtained from D by replacing each nonzero entry of D by a 1, where the rows and columns of $D^\#$ inherit their labels from D . $[I_r \mid D^\#]$ is sometimes called a *partial representation* for M because of the following result ([Oxl11, 6.4.1]):

Proposition 6.1.1. *Let $[I_r \mid D_1]$ and $[I_r \mid D_2]$ be matrices over the fields \mathbb{K}_1 and \mathbb{K}_2 with the columns of each labeled, in order, by e_1, e_2, \dots, e_n . If the identity map on $\{e_1, e_2, \dots, e_n\}$ is an isomorphism from $M[I_r \mid D_1]$ to $M[I_r \mid D_2]$, then $D_1^\# = D_2^\#$.*

Let $G(D^\#)$ denote the associated simple bipartite graph, that is, $G(D^\#)$ has vertex classes $\{e_1, e_2, \dots, e_r\}$ and $\{e_{r+1}, e_{r+2}, \dots, e_n\}$, and two vertices e_i and e_j are adjacent if and only if $D_{e_i, e_j}^\# = 1$. Equivalently, $G(D^\#)$ is the fundamental graph $G_B(M)$ associated with the basis $B = \{e_1, e_2, \dots, e_r\}$ of M (See [Oxl11, 4.3.2]).

6.2. Brylawski-Lucas Theorem. The following theorem gives a best-possible lower bound on the number of entries of D whose values can be predetermined by row and column scaling, and also provides an easy way to see whether or not two representations are projectively equivalent.

Theorem 6.2.1 (Brylawski and Lucas 1976). *For a field \mathbb{F} , let the $r \times n$ matrix $[I_r \mid D_1]$ be an \mathbb{F} -representation of a matroid M . Let $\{b_1, b_2, \dots, b_k\}$ be a basis of the cycle matroid of $G(D_1^\#)$. Then*

- (1) $k = n - \omega(G(D_1^\#))$.
- (2) *If $(\theta_1, \theta_2, \dots, \theta_k)$ is an ordered k -tuple of nonzero elements of \mathbb{F} , then M has a unique \mathbb{F} -representation $[I_r \mid D_2]$ that is projectively equivalent to $[I_r \mid D_1]$ such that, for each $i \in \{1, 2, \dots, k\}$, the entry of D_2 corresponding to b_i is θ_i . Indeed, $[I_r \mid D_2]$ can be obtained from $[I_r \mid D_1]$ by a sequence of row and column scalings.*

We repeat the proof of Theorem 6.2.1 from [Oxl11, 6.4.7] here, in order to contrast with that of Lemma 6.3.1. We omit the proof that $[I_r \mid D_2]$ is unique, and refer the reader to [Oxl11, 6.4.7].

Proof. (1) follows from the fact that, if $M = M(G)$ where G is a graph, then $r(M) = |V(G)| - \omega(G)$.

To prove (2), let $G_1 \subseteq G(D_1^\#)$ be a forest, and for each edge $x \in E(G_1)$, $\theta(x)$ is a nonzero element of \mathbb{F} . We will show by induction on $|E(G_1)|$ that, by a sequence of row and column scalings of

$[I_r \mid D_1]$, we can obtain a matrix $[I_r \mid D_2]$ such that, for all edges $x \in E(G_1)$, the entry in D_2 corresponding to x is $\theta(x)$.

The hypothesis is trivially true for $|E(G_1)| = 0$. Assume it is true for $|E(G_1)| < m$ and let $|E(G_1)| = m \geq 1$. As G_1 is a forest with at least one edge, it has a vertex v of degree one. Let $y \in E(G_1)$ be the edge incident with v . By induction, there is a sequence of row and column scalings of $[I_r \mid D_1]$ to obtain $[I_r \mid D_2]$ such that, for all edges $x \in E(G_1 \setminus y)$, the entry of D_2 corresponding to x is $\theta(x)$.

Now v corresponds to a row or a column of D_2 . If v corresponds to a column, then we can scale that column to make the y -entry equal to $\theta(y)$ without altering any of the $(G_1 \setminus y)$ -entries. If v corresponds to a row, then as y is the only edge meeting v , none of the entries of D_2 corresponding to edges of $G_1 \setminus y$ is in this row. So we can multiply this row by an appropriate nonzero scalar t to make the y -entry equal to $\theta(y)$ without changing any of the $(G_1 \setminus y)$ -entries. This multiplication may alter the entry in row v of I_r , but we can fix this by multiplying the corresponding column of I_r by t^{-1} without affecting any other entries.

We conclude by induction that there is a sequence of row and column scalings of $[I_r \mid D_1]$ to obtain $[I_r \mid D_2]$ in the desired form. \square

6.2.1. *Example.* Let $B = \{e_1, e_2, e_3\}$. For the non-Fano F_7^- , the B -fundamental-circuit incidence matrix over a field \mathbb{F} of characteristic $\neq 2$ is given below:

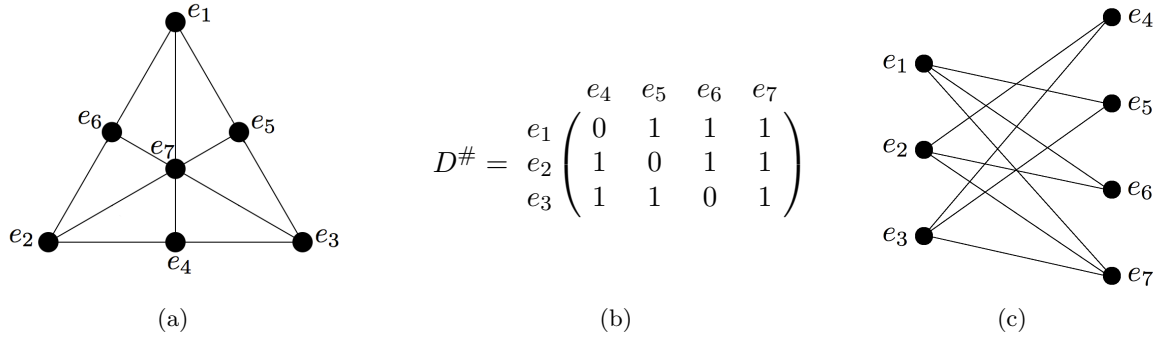


FIGURE 8. (a) A geometric representation for F_7 , (b) the B -fundamental-circuit incidence matrix, and (c) its associated bipartite graph.

The boxed entries in $D^\#$ below correspond to a basis of the cycle matroid of $G(D^\#)$.

$$D^\# = \begin{matrix} & \begin{matrix} e_4 & e_5 & e_6 & e_7 \end{matrix} \\ \begin{matrix} e_1 \\ e_2 \\ e_3 \end{matrix} & \begin{pmatrix} 0 & \boxed{1} & \boxed{1} & \boxed{1} \\ \boxed{1} & 0 & 1 & \boxed{1} \\ 1 & \boxed{1} & 0 & 1 \end{pmatrix} \end{matrix}$$

By Theorem 6.2.1, we can independently assign arbitrary nonzero elements of \mathbb{F} to each of those boxed entries. Note that $n - \omega(G(D^\#)) = 7 - 1 = 6$.

6.3. Restricting column scaling. If we restrict column scaling factors to ± 1 , then we cannot choose $\theta(x)$ freely as we did in Theorem 6.2.1, and instead we are only restricted to $\theta(x)$ such that the entry corresponding to x is $\pm\theta(x)$. The following lemma illustrates this.

Lemma 6.3.1. *With notation as in Theorem 6.2.1, let $(\phi_1, \phi_2, \dots, \phi_k)$ be an ordered k -tuple such that the entry of D_1 corresponding to b_i is $\pm\phi_i$. Let $[I_r \mid D']$ be the matrix such that, for each $i \in \{1, 2, \dots, k\}$, the entry of D' corresponding to b_i is ϕ_i . Then we can obtain $[I_r \mid D']$ from $[I_r \mid D_1]$ by a sequence of row and column scalings by ± 1 .*

Proof. Let G_1, D_2, v , and y be as in the proof of Theorem 6.2.1. Again, v can label a row or a column of D_2 .

If v corresponds to a column, then we can multiply the column by ± 1 to make the y -entry equal to $\phi(y)$ without altering any of the $(G_1 \setminus y)$ -entries. If v corresponds to a row, then we can multiply this row by an appropriate nonzero scalar t to make the y -entry equal to $\phi(y)$ without changing any of the $(G_1 \setminus y)$ -entries. However, this multiplication may alter the entry in row v of I_r . Since we only allow column scaling by -1 , we can only fix this (by multiplying the corresponding column of I_r by ± 1) if $t = \pm 1$.

We note that in either case, $\phi(y)$ can only be ± 1 times the original entry of D_1 corresponding to y , because column scaling is restricted to scaling by ± 1 . \square

7. CONCLUSION

We have shown that whether to allow column scaling in relating two signed-graphic representations is a relevant question. Our examples show that column scaling is sometimes necessary in order to transform one signed-graphic representation into another; moreover, there exist many collections of signed-graphic representations that row-reduce to the same standard form.

Given a binet matrix which can be obtained by row-reducing several different signed graphs, Musitelli does not mention which of the signed graphs is found by his algorithm, and his algorithm does not find all the signed graphs. It is possible that his algorithm makes some consistent canonical choice, or finds certain representations more attractive than others, but this is not mentioned in his paper. Because Musitelli's paper ignores the existence of both other row-equivalent and other row-inequivalent representations, there is some doubt as to what his algorithm does.

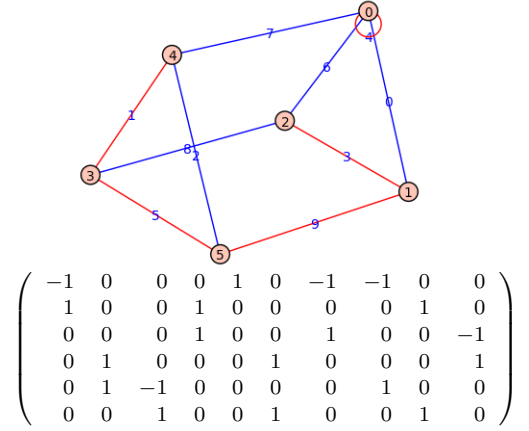
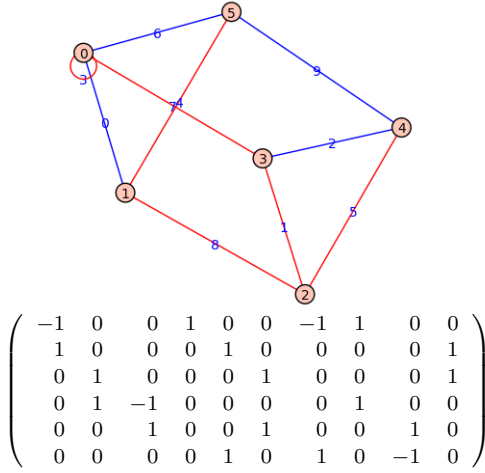
APPENDIX A. EXAMPLES OF SIGNED-GRAPHIC REPRESENTATIONS

A.1. A matroid with some row-equivalent, some non-row-equivalent signed-graphic representations. The following dyadic matroid of rank 6 on 10 elements whose \mathbb{D} -representation is

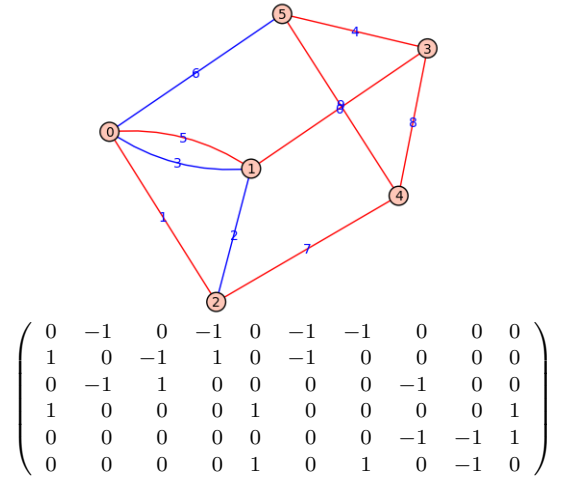
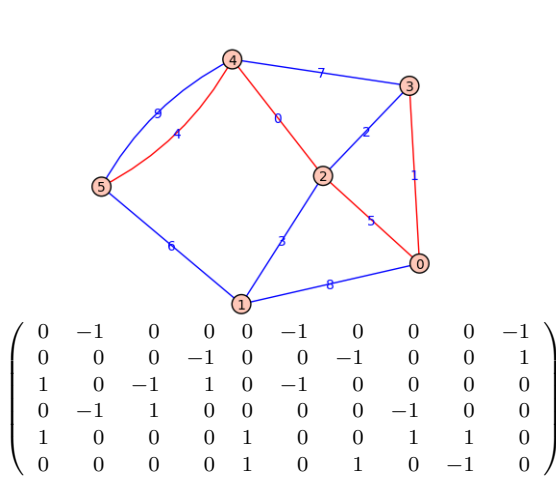
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 \end{pmatrix}$$

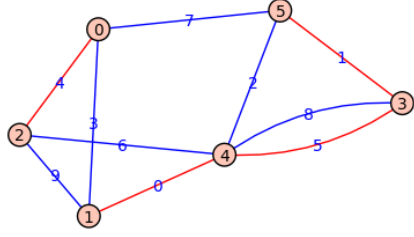
has 11 signed-graphic representations, some of which are row-equivalent and some of which are not.

The following two signed-graphic representations are not row-equivalent to any of the other representations; in other words, column scaling is required in order to transform either representation into a different signed-graphic representation:

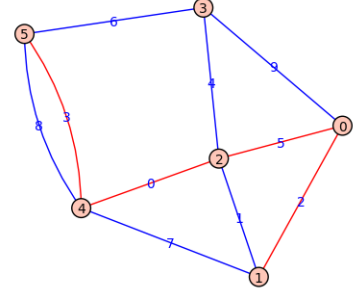


The remaining representations are all row-equivalent to one another:

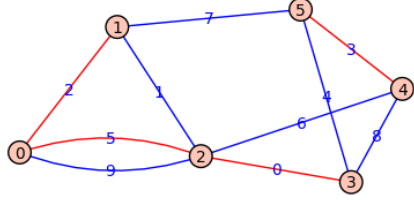




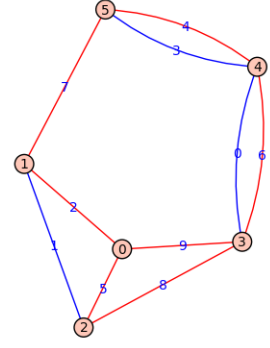
$$\begin{pmatrix} 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$



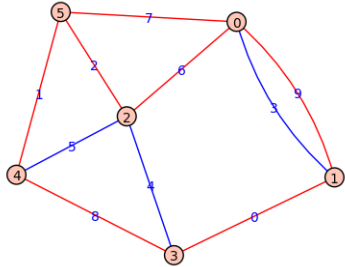
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}$$



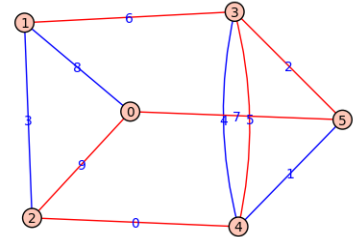
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$



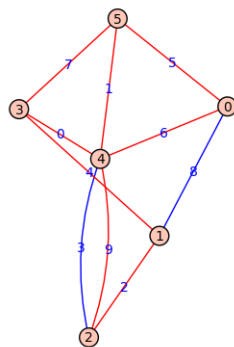
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \\ -1 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

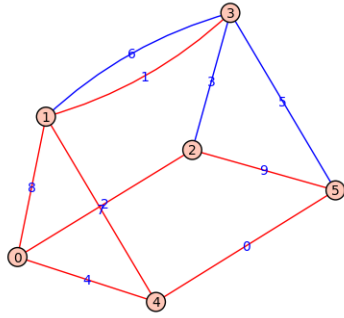


$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 0 & 1 \end{pmatrix}$$

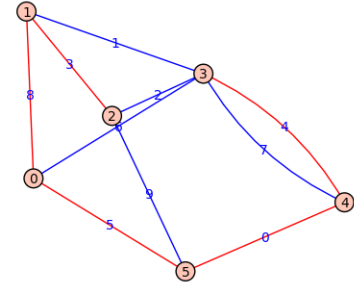
A.2. A matroid with all row-equivalent signed-graphic representations. The following dyadic matroid of rank 6 on 10 elements whose \mathbb{D} -representation is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 1 \end{pmatrix}$$

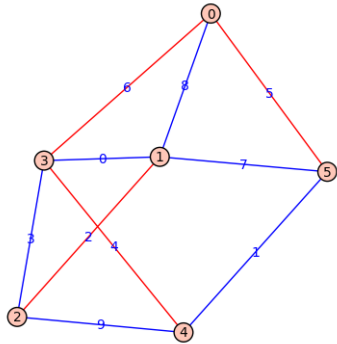
has 15 signed-graphic representations, every pair of which is row-equivalent:



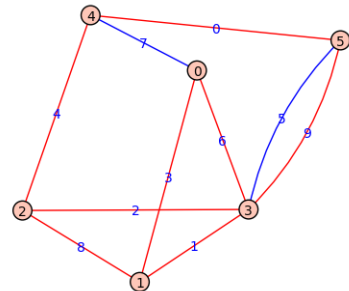
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



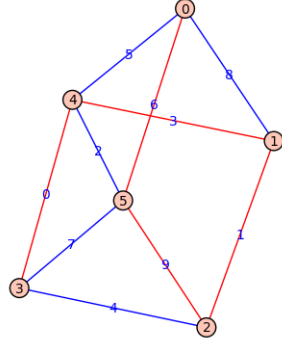
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



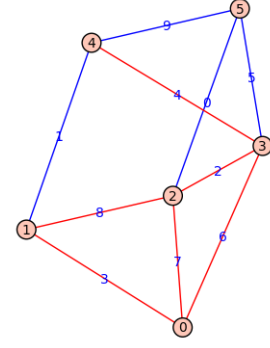
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \end{pmatrix}$$



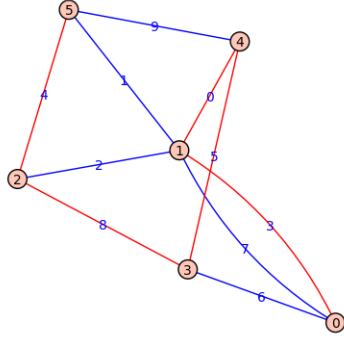
$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



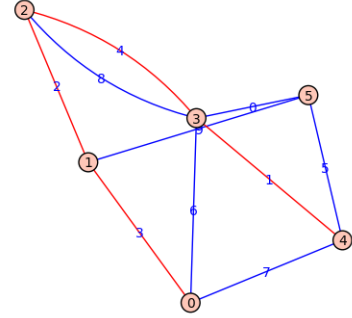
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$



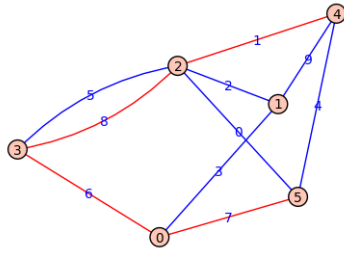
$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

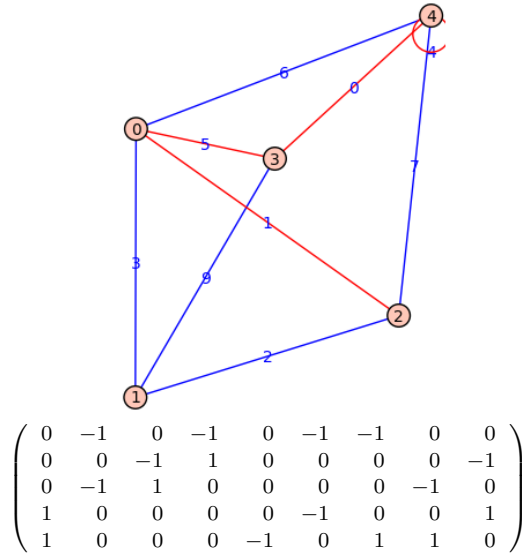
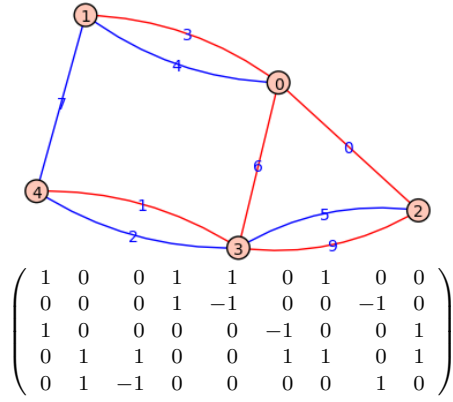
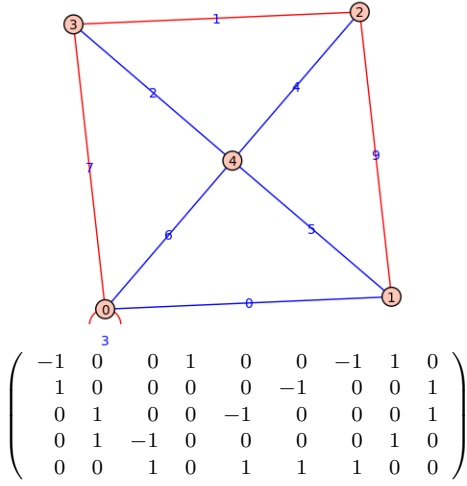


$$\begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

A.3. A matroid with all non-row-equivalent signed-graphic representations. The following dyadic matroid of rank 5 on 9 elements whose \mathbb{D} -representation is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 1 \end{pmatrix}$$

has 3 signed-graphic representations, every pair of which is not row-equivalent:



ACKNOWLEDGEMENTS

I would like to express my very great appreciation to Professor Stefan van Zwam, my research advisor, for teaching me about matroid theory from the ground up, and for his patient guidance and valuable suggestions during the planning and development of this research work. I would also like to thank Professor Rudi Pendavingh for his algorithm that allowed us to generate all signed-graphic representations of a given dyadic matroid.

REFERENCES

- [Cam06] P. Camion, Unimodular module, Discrete Mathematics, 2006.
- [Mus10] A. Musitelli, Recognizing binet matrices, Ph.D Thesis, 2010.
- [Oxl11] J. G. Oxley, *Matroid Theory*, Oxford University Press, New York, 2011.
- [Piv11] I. Pivotto, Even cycle and even cut matroids, Ph.D Thesis, University of Waterloo, 2011.
- [S⁺09] W.A. Stein et al., *Sage Mathematics Software (Version 6.1.1)*, The Sage Development Team, 2014, <http://www.sagemath.org>.
- [Zwa09] S.H.M. van Zwam, Partial fields in matroid theory, Ph.D. thesis, Technische Universiteit Eindhoven, 2009.
- [PZ07] R.A. Pendavingh and S.H.M. van Zwam, $\sqrt[3]{1}$ -signed graphs, 2007.
- [PZ07a] R.A. Pendavingh and S.H.M. van Zwam, Skew partial fields, multilinear representations of matroids, and a matrix tree theorem, 2007.

E-mail address: lil@princeton.edu